1. DATABASE CONCEPTS

Section 1 presents an overview of the SAPHIRE database structure. Included in this section are discussions of SAPHIRE projects, base case versus current case, base case updates, and change sets.

1.1 SAPHIRE Projects

♦ In SAPHIRE, the term "project" represents a single, specific data base.

♦ When starting a PRA, a project must be created (via the **File ➔ New Project** option).

① Click the backup up one level folder or use the menu bar to select the directory where the new project is to be added.

② Right click the mouse and scroll down to **new ➔ folder**

③ Add the new folder (name) directory for the project.

④ Select the newly created project folder

⑤ In the file name field, type in the project name (up to 24 characters).

⑥ The project can now be modified using (**Modify ➔ Project**).

♦ To access a particular project, the project must be selected (via the **File ➔ Open Project** option).

◊ Modifications to a data base (e.g., a new fault tree is developed) are always made to the currently selected project.

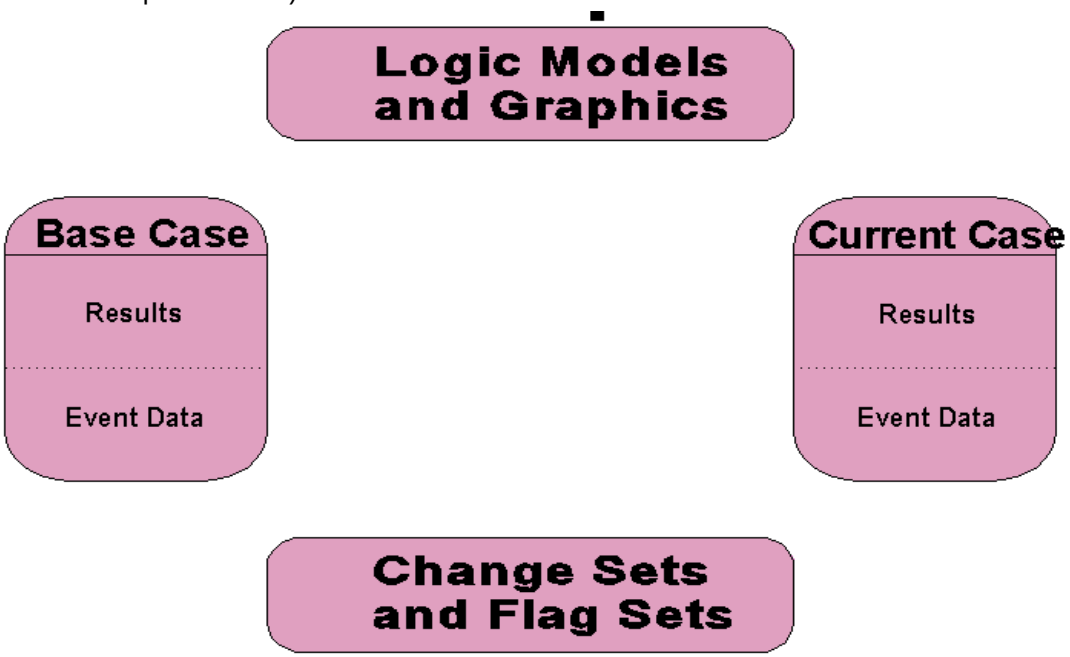
◊ For a given project, only one list is kept for all types of information. Thus, within a project, only a single copy of a particular fault tree, event tree, or basic event is ever stored in the data base.

Project (Definition)

A group of fault tree logic and graphics; event trees and sequences; basic events and related data; cut sets; analysis results; and descriptions.

1.2 Base Case Versus Current Case Data

- Base case and current case are two separate parts of a project data base.
  - ◊ **Base Case** data is stored in the data base files as a "permanent" record
  - ◊ **Current Case** data is used to perform an analysis (e.g., cut set generation and quantification)



Current Case Is

- ◊ Created (via the **Generate** option) by applying change sets to base case data
- ◊ Used for sensitivity or event analysis
- ◊ Not stored in relational database (only the change set and base case data are stored)

- All SAPHIRE calculations use the data stored in the **current** case.

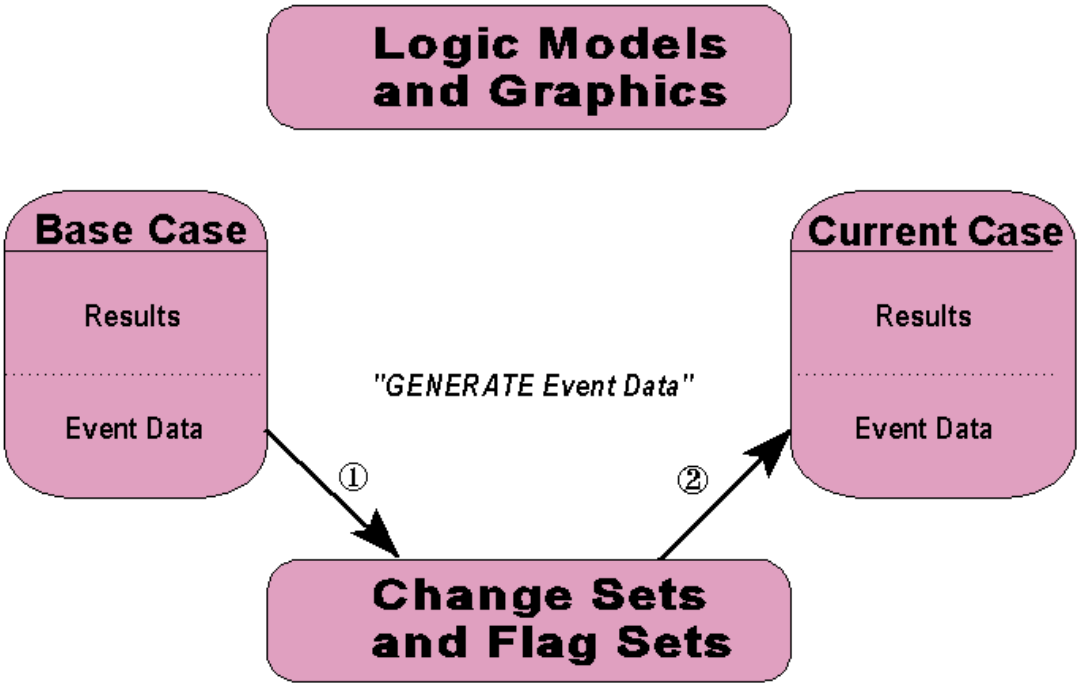
- Current case is sometimes referred to as the Alternate case.

- Current case can equal the base case in order to reproduce the original study stored in the base case.

- Only current case cut sets can be modified with the cut set editor. Changes made to cut sets with the cut set editor are implemented automatically (no cut set update is necessary).

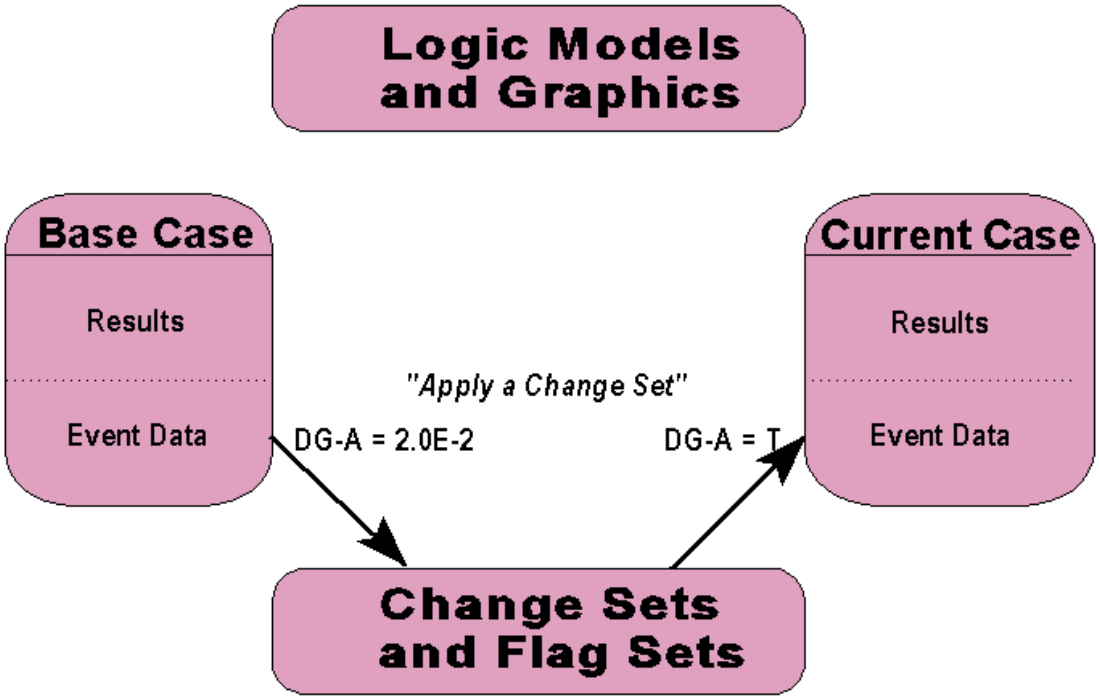
1.3 Generating Event Data

Generate option transfers base case data to the current case (after making changes specified in any marked change sets). SAPHIRE always uses current case data for analysis. If the base case data is changed and the **Generate** option is *not* performed, the data that is used for the analysis will not reflect the changes.



Change Sets (Definition)

Change Sets are a user-defined set of changes that will be applied to the base case data when event data is transferred to the current case (via the Generate option). Multiple change sets can be defined and applied singly or in combination.



1.4 Rules for Creating and Using Change Sets

- No limit to the number of change sets that can be added to the data base
- Change set name is limited to 24 characters, the description is limited to 60 characters
- A change set can contain one class change and unlimited individual probability changes.
- Multiple change sets can be used in combination to create different sensitivity studies.

EXAMPLE: Two change sets are developed. The first is named "A" and sets all valves to failed. The second is named "B" and sets all pumps to failed. The possible scenarios are

Change set(s) that are marked	Sensitivity case
None	Original base case analysis
A	Valves failed
B	Pumps failed
A and B	Both valves and pumps failed

- The order of "marking" a change set is important. (Change sets are marked by double-clicking the line containing the change set.)
-
  - The first selected change set will be the first one that is applied
  - Later changes will overwrite earlier ones if there is any overlap
  - Within a change set, individual probability changes will overwrite a class change

- Base case data and changes made to the current case can be viewed by using the **Generate ➔ Report** option.

  - Unaffected events *(those events not modified by a change set)*
  - Affected events *(those events which are modified by a change set)*
  - All events *(the current probability for all events)*

Class Changes

Class changes use a basic event attribute to search for a class of basic events to which the defined change applies

- The search criteria are defined first
- The change to be applied is then defined

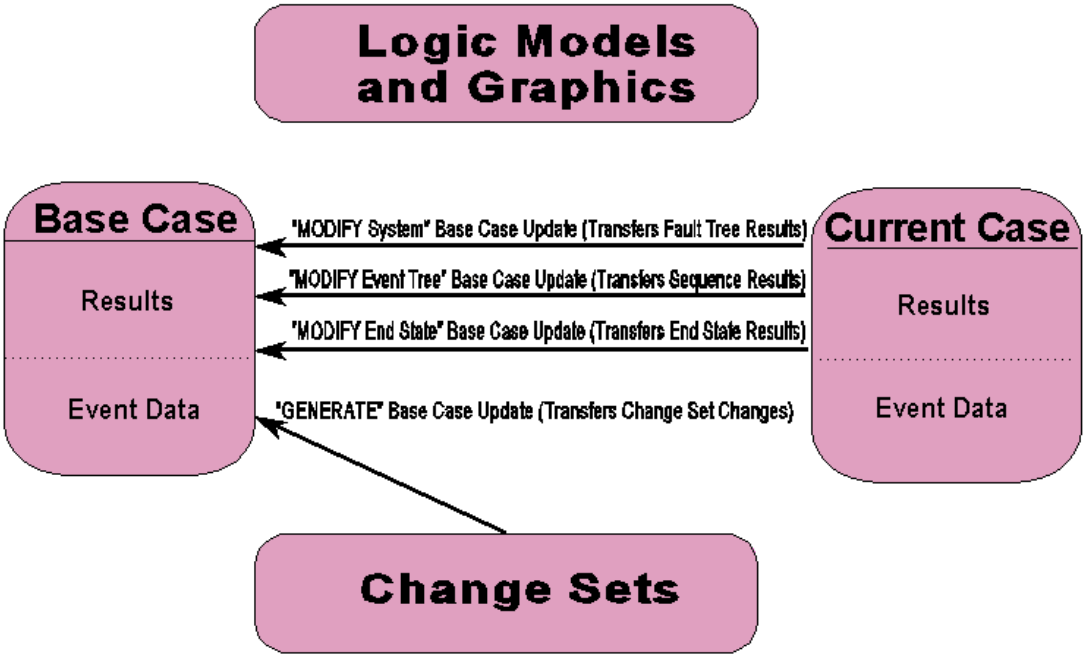
Single (a.k.a. Probability) Changes

Single changes only modify individual, user-identified basic events

- The desired basic event is selected
- The changes to the basic event are then defined

1.5 Base Case Update

- Base case data and results are changed by **updating** the base case. Updating the base case transfers the current case data or results into the base case.
  - **Base Case** results are stored in the data base files as a “permanent” record

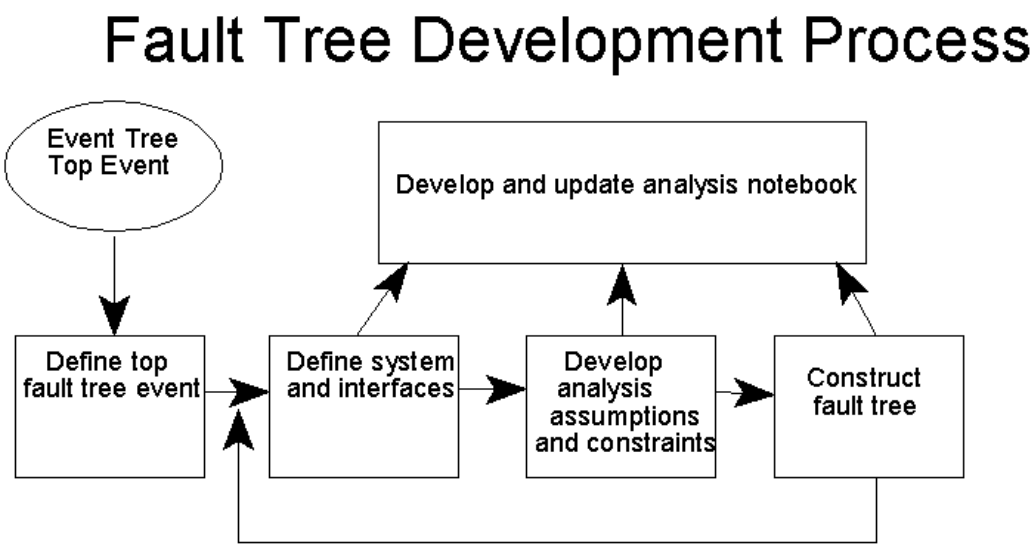


2. BUILDING FAULT TREES

Section 2 introduces fault tree development, SAPHIRE fault tree symbols, and SAPHIRE fault tree modeling conventions. You will learn how to enter and edit fault trees by using the fault tree text editor and the graphical fault tree editor.

2.1 Fault Tree Development

Definition: Fault tree analysis is a deductive failure analysis method which focuses on identifying all of the credible ways that can cause an undesired event to occur. The undesired event is stated at the top of the fault tree. The fault tree gates specify the logical combinations of basic events that lead to the top event.



2.2 Basic Event Symbols

Basic Event

Boxed Basic Event

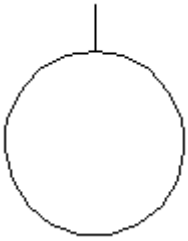
Table of Basic Events

Undeveloped Event

House Event

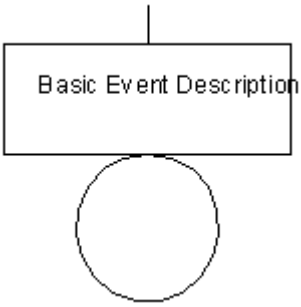
Undeveloped Transfer

♦ Basic Event



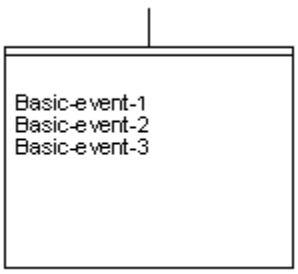
A basic event represents a fault such as a hardware failure, human error, or an adverse condition. The circle signifies that the fault event does not require further development.

♦ Boxed Basic Event



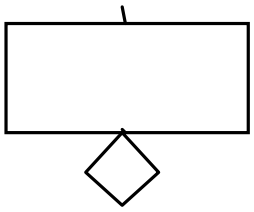
An alternate symbol for a basic event is a boxed basic event that provides a box to contain the description of the basic event.

♦ Table of Basic Events



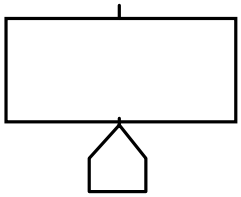
The table of basic events symbol allows up to 8 basic events to be entered in a space-saving layout. The logic used by the table is dictated by the gate it is connected to in the fault tree.

♦ **Undeveloped Event**



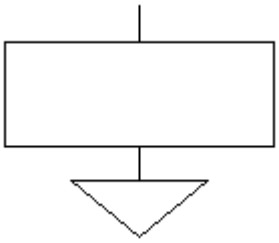
The undeveloped event denotes a basic event that is actually a more complex event that has not been further developed by fault tree logic. SAPHIRE treats this event no differently than a basic event.

♦ **House Event**



The house event is used to denote a failure that is guaranteed to always occur or never to occur. However, the calculation type assigned to a basic event establishes whether or not an event is a house event. Consequently, any basic event in SAPHIRE can be made into a house event.

♦ **Undeveloped Transfer**



The undeveloped transfer indicates that the event is complex enough to have its own fault tree logic developed elsewhere; however, the event has been treated as a basic event in the present fault tree.

2.3 Logic Gate Symbols

AND Gate

OR Gate

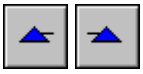
N/M Gate

Transfer Gate

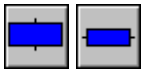
Inhibit Gate

NOT AND (NAND) Gate

NOT OR (NOR) Gate



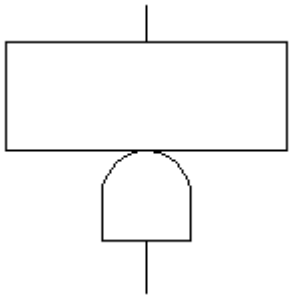
Right and Left Transfer



Vertical and Horizontal Boxes

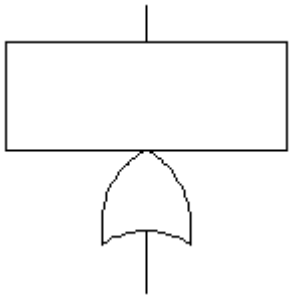
Connecting Lines

♦ **AND Gate**



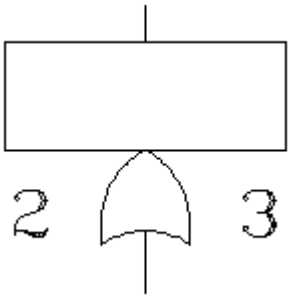
All inputs to the AND gate must occur for failure to occur.

♦ **OR Gate**



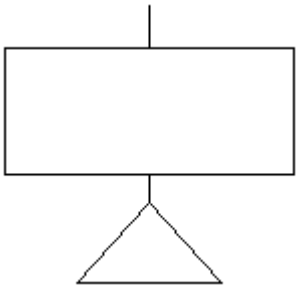
Any one input to the OR gate will cause failure to occur.

♦ **N/M Gate**



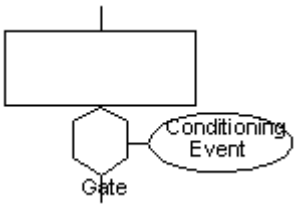
This gate states that N of the M input events must occur for failure to occur. For a 2/3 gate, any combination of 2 of the 3 input events must occur.

♦ **Transfer Gate**



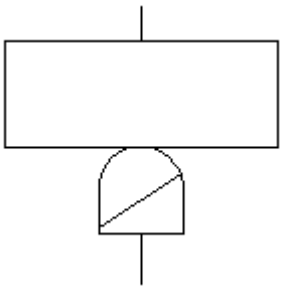
This gate is used to link logic structures together without introducing any new logic of its own. The transfer gate indicates that logic is continued on a new page (or on the same page). The transfer gate name is the same as the gate where the logic continues, and when transferring to another page (a separate fault tree file), the gate being transferred to must be the top gate on the page. (The top gate name of a fault tree must always be the fault tree filename.)

♦ **Inhibit Gate**



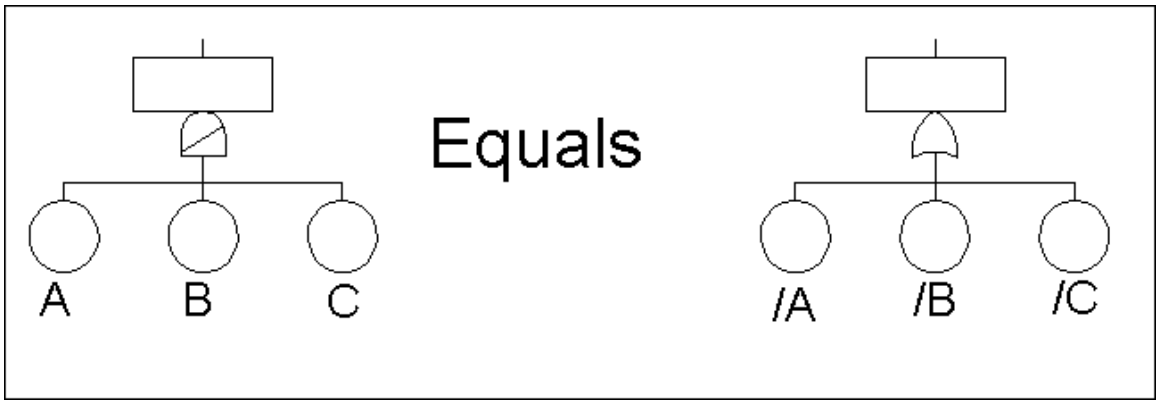
The output occurs if the single input fault occurs in the presence of an enabling condition. Thus, the inhibit gate is a special type of AND gate. The enabling condition (or conditioning event) is treated simply as a basic event with a probability or as a house event.

♦ **NOT AND (NAND) Gate**

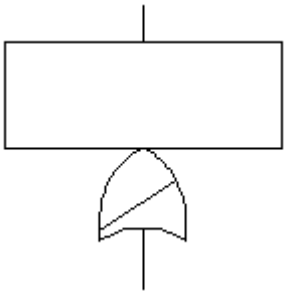


The output occurs if any one of the inputs does not occur. (See example.)

♦ **NAND Gate Example**

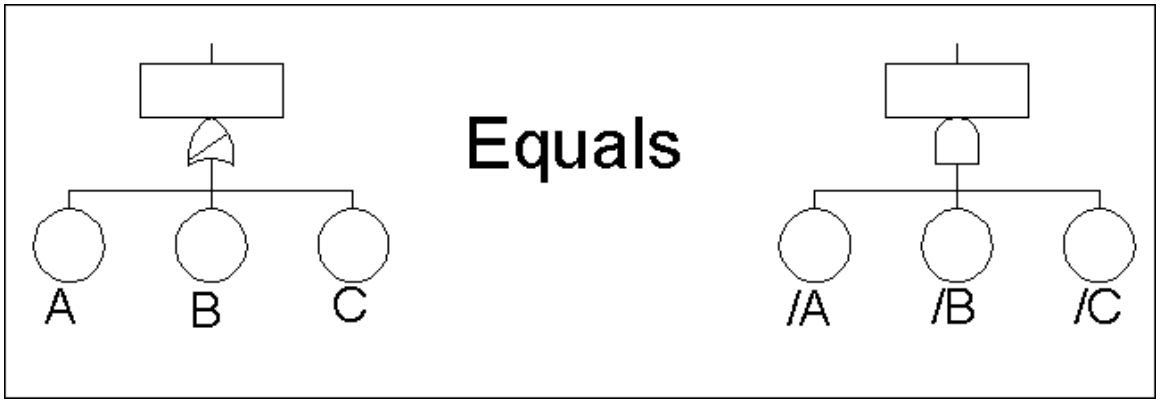


♦ **NOT OR (NOR) Gate**

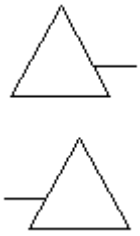


The output occurs if none of the inputs occur. (See example.)

♦ **NOR Gate Example**

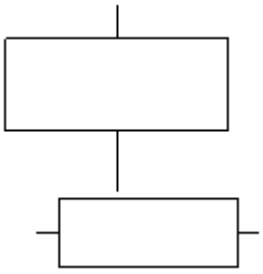


♦ **Right Transfer and Left Transfer**



These symbols are used to indicate where a transfer has taken place.

♦ **Vertical and Horizontal Boxes**



A vertical or horizontal box is provided to allow further descriptive information to be placed in the diagram. SAPHIRE ignores these boxes when processing the fault tree.

♦ **Connecting Lines**

Connecting lines can be solid or dashed, or a dotted/dashed line. Connecting lines can be drawn at any angle. The connecting lines must actually touch the symbols being connected at the input or output stem on the symbol.

2.4 SAPHIRE Fault Tree Conventions

♦ Fault Tree File Name

The fault tree name **must** be the same as the top gate name. The fault tree name can be 24 characters long.

♦ Fault Tree Gates

24 characters allowed.

♦ Basic Event Names

24 characters allowed.

♦ Top Gates

A fault tree "page" or file can have only one top gate.

♦ SAPHIRE Default Naming

SAPHIRE will automatically assign basic event names (EVENTn) and gate names (GATEn); however, the user may replace the default name. These defaults may changed on the *Preferences* dialog.

♦ Transfer Fault Trees

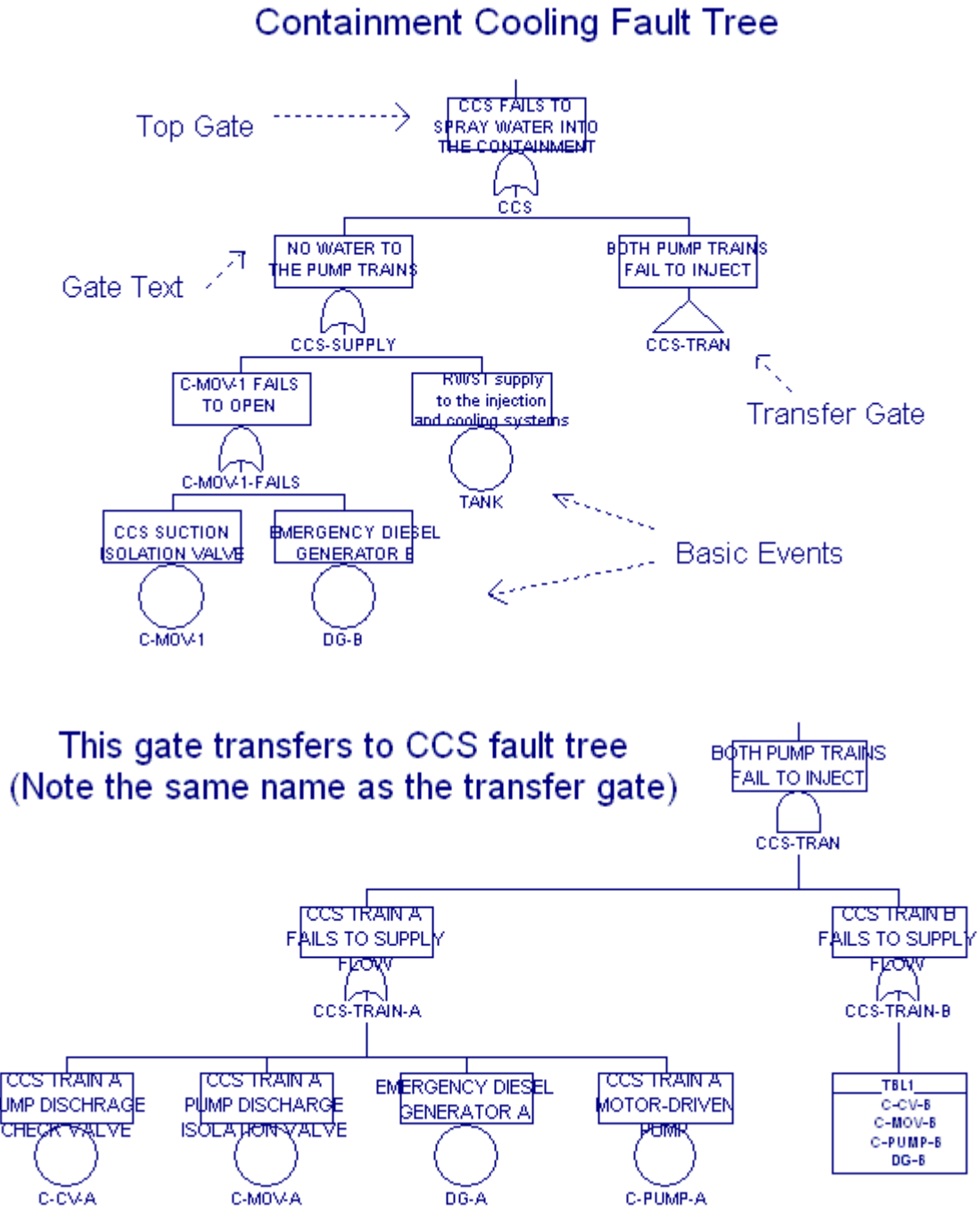
A fault tree can transfer to another fault tree by using transfer gates. The transfer gate name specifies the gate to transfer to. The transfer gate name must be the same as the gate name being transferred to. The gate being transferred to must either (1) be on the same page or (2) be the top gate of a separate fault tree file.

♦ Complemented Events

Complemented events can be input into fault trees by entering the basic event name with the "/" symbol, e.g., /DG-A.

- ◊ Do not use *, ?, \, @, /, or space in the naming SAPHIRE fault trees or basic events. Note that the "/" symbol is used to denote an complemented event.

5.5 Example Fault Tree

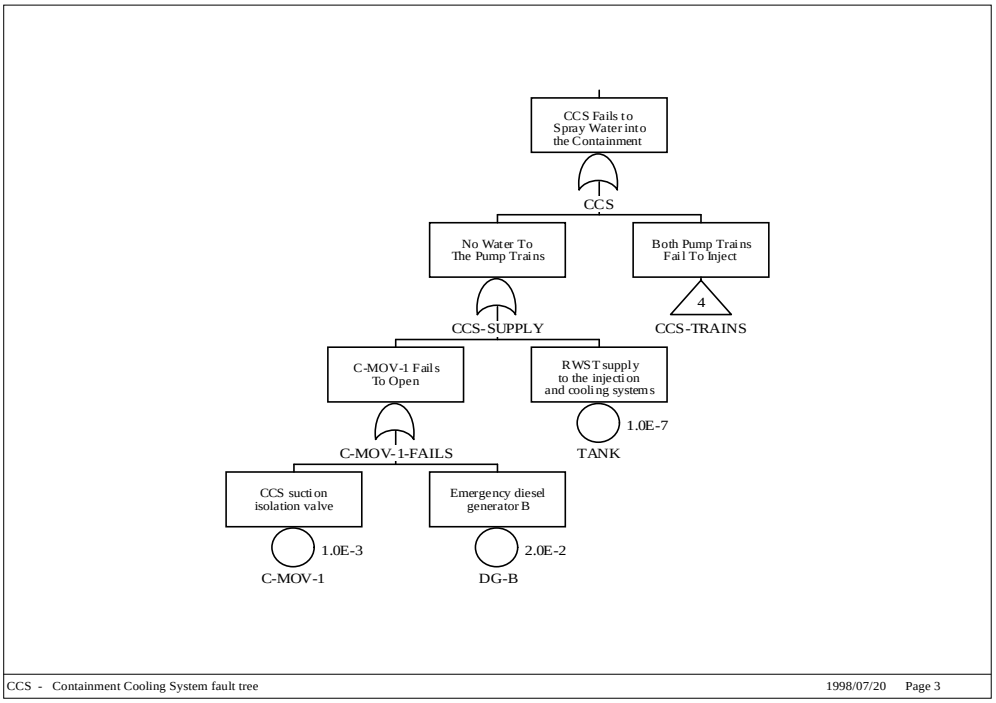


2.6 Fault Tree Logic Editor

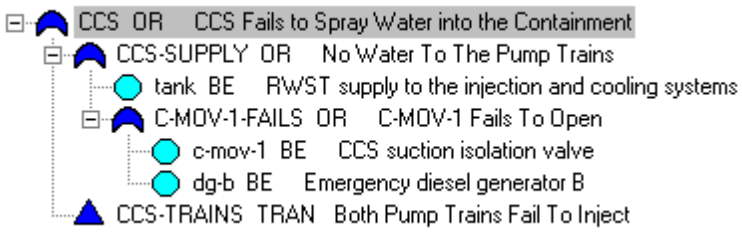
The SAPHIRE fault tree logic editor allows you to modify the logic of a SAPHIRE fault tree (system or subsystem). Fault tree gate descriptions can be entered from this editor as well as basic event descriptions and data. The fault tree logic editor is a convenient tool for quickly editing fault tree logic and entering gate and basic event data.

Shown below are logical representations of both the fault tree graphic and fault tree logic.

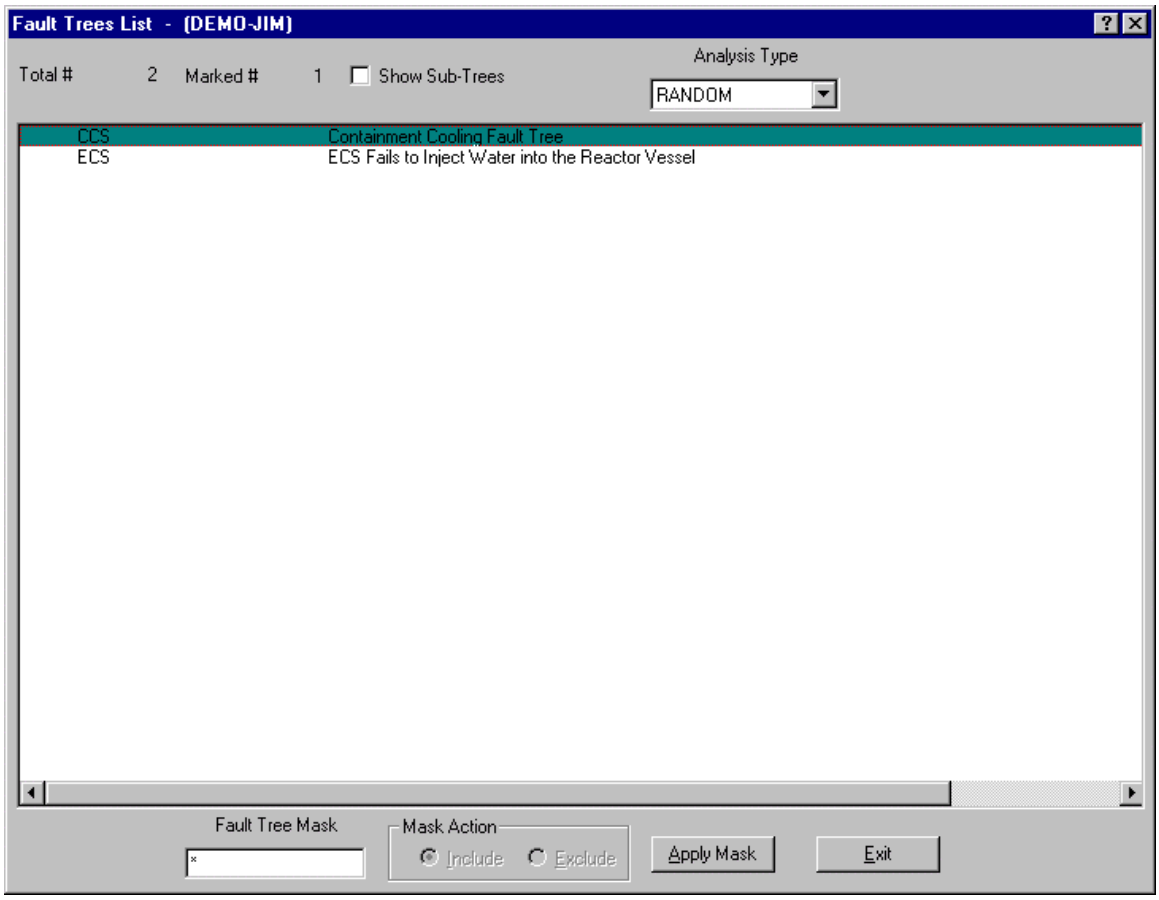
Interpreting Fault Tree Logic Format



Logic editor format



The Fault Tree Logic Editor



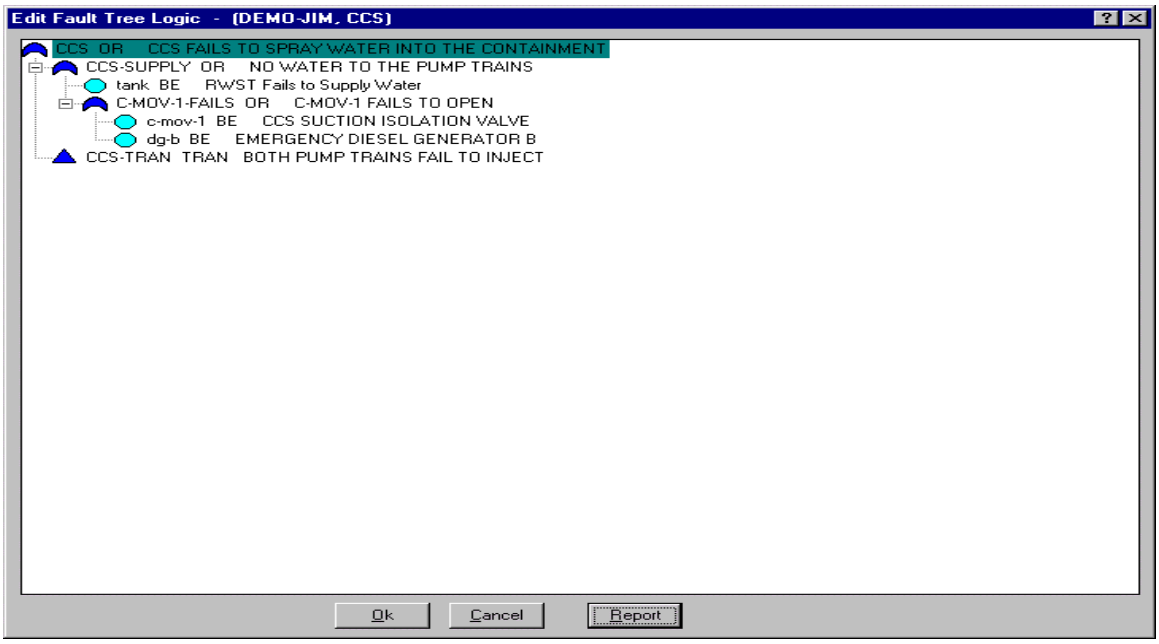
- Select the **Fault Tree** menu.

- When you select this menu, the *Fault Tree List* dialog is displayed. The fault trees are shown in the list box. To display both systems *and subsystems*, select the Sub-systems check box. (Deselecting the Sub-systems check box will toggle the list so that only systems are displayed.)

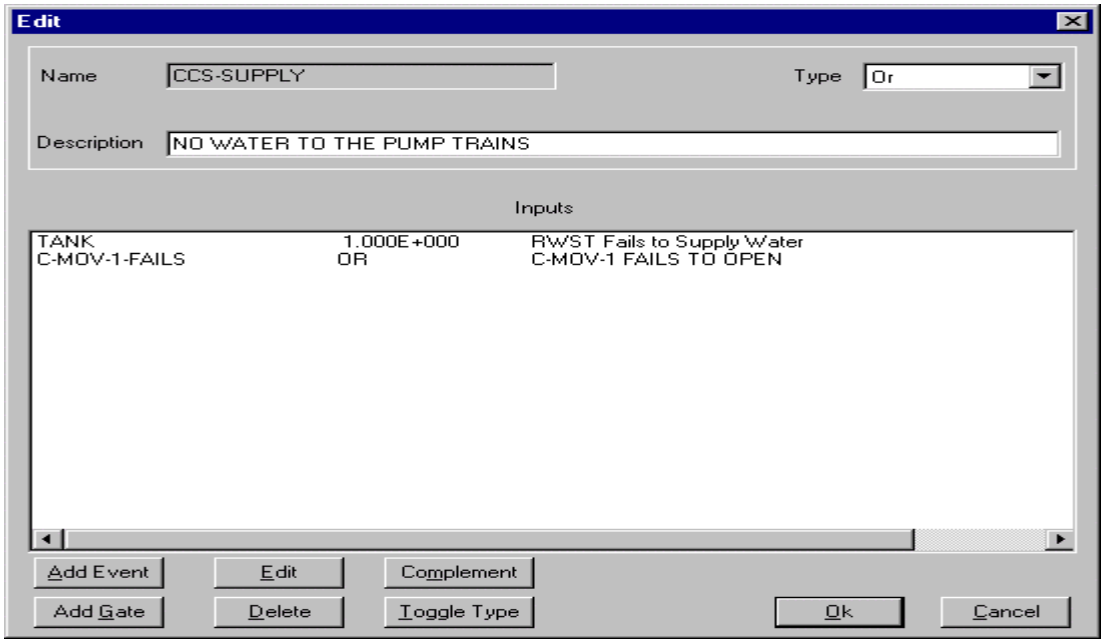
- To edit a fault tree, highlight the fault tree and select Edit Logic from the pop-up menu. (Right-click to invoke the pop-up menu.)

- Fault trees are added to the database by adding tops to an event tree, by using **Modify ➔ Fault Tree ➔ Add**, or by **Fault Tree ➔ Add Fault Tree**.

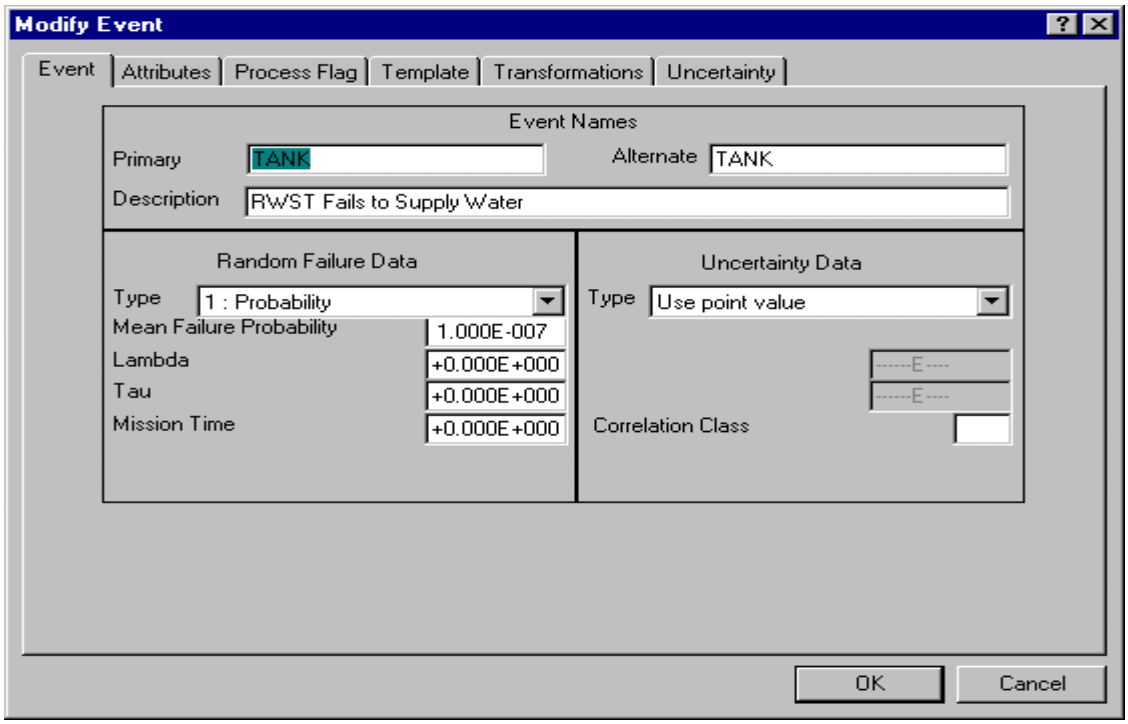
The Logic Editor Display Screen



- The logic editor uses a hierarchical approach to editing the fault tree.
- The logic can be expanded by pressing + or collapsed by pressing -.
- To modify a basic event (or gate), highlight the gate it inputs to (click the right mouse) and select modify

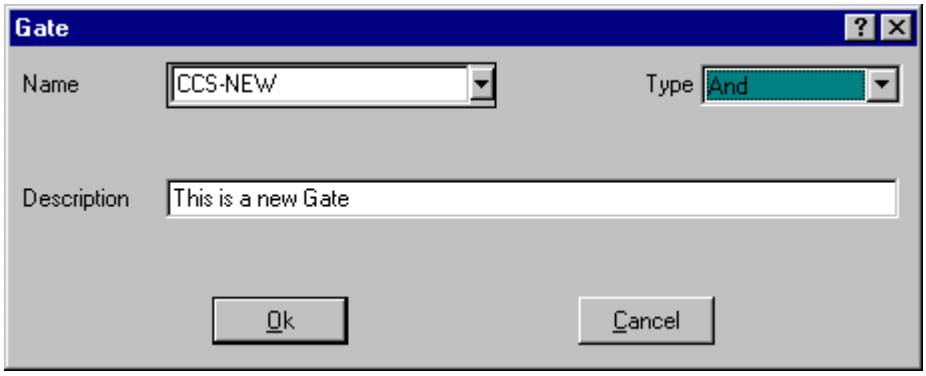


- Highlight the basic event (or gate) and select **Edit**. Then modify its name, description, or probability.



- To add a gate (or basic event), highlight the gate it inputs to (click the right mouse) and select modify.

- Select the **Add Gate** option and type in the name of the new gate along with its type and description.



The following provides a list of the options available for adding or modifying both basic events or gates.

Add Event – Add event is used to add a new basic event as an input to the gate. This option allows for the basic event's description to be added along with its probability.

Add Gate – Add gate is used to add a new gate as an input to the gate highlighted. This option also allows for the gate's description to be added.

Edit – The edit option allows you to change a basic events name, description, and probability and change a gate's name, type, or description depending on which type of event is highlighted.

Delete – The delete option will delete the highlighted basic event or gate.

Complement – The complement option will complement any basic event that is highlighted.

Toggle Type – Toggle type will toggle the event highlighted between a basic event or a gate.

Ok – Close the Modify (Edit) fault tree dialog and apply the changes.

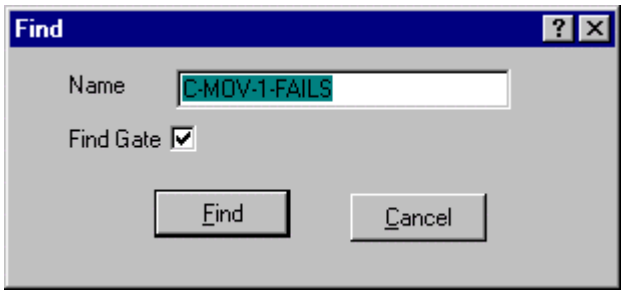
Cancel – Close the Modify (Edit) fault tree dialog without applying the changes.

Other Features of the Logic Editor

Find Option

- The find option searches the fault tree logic to find the specified basic event or gate.

- To find a basic event or gate (right click the mouse) and select **Find**. Type in the name of the gate or basic event and check the box if the search is for a gate and uncheck the box if the search is for a basic event. Press **Find**.



Move Option

- The move option will move a gate and its logic from its original input gate to another selected gate.

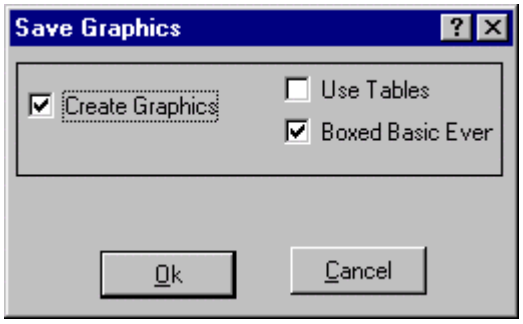
- To move a gate highlight the gate (right click the mouse), select move and place cursor by the new gate and click the left mouse button. The gate and its associated logic will now be placed as an input to the new gate.

Transfer Option

- The transfer option transfers from the fault tree currently being edited to the fault tree specified by the transfer gate.
- To transfer from one fault tree to another highlight the transfer gate (right click the mouse) then select **Follow Transfer**.
- To transfer back, highlight the transfer gate (right click the mouse) then select **Transfer Back**.

2.7 Updating the Graphical Fault Tree (.DLS) File

After changes are made to fault tree logic using the Logic Editor, when you press **Ok** a dialog box will ask if you what to save the fault tree logic as a graphic file. At this point you can check the box and a graphic file will automatically be created. Or secondly, use the **Utility ➔ Fault Tree ➔ Alpha-to-Graphics** conversion to update the fault tree graphical file. (You do not have to update the graphical file in order to perform analyses with the updated logic; however, to keep the fault tree logic and the fault tree graphic identical, you must perform the Alpha-to-Graphics conversion.)



2.8 Fault Tree Editor Tips

- New fault tree "systems" cannot be added using the logic editor. Systems can be added by saving the new fault tree in the Fault Tree graphical editor, by entering the fault tree name as the top event in an event tree, or by adding it in **Modify ➔ Fault Tree**, or by adding it in **Fault Tree ➔ Add Fault Tree**. Subtrees are added in the editor when the transfer gate type is used.

- The fault tree logic editor does not allow the user to save the edited fault tree with a new fault tree name. The top gate in the fault tree should match the fault tree name.

2.9 Fault Tree Graphics Features Guide

Beginning The Editing Session

- To develop a new fault tree

  SAPHIRE menu bar: **Fault Tree**

  Fault Tree List dialog pop-up menu: **Edit Graphics**

  Fault Tree Editor menu bar: **File ➔ New**

  or

  SAPHIRE menu bar: **Fault Tree**

  Fault Tree List dialog pop-up menu: Add Fault Trees

- If you wish to edit an existing file while in the graphic editor

  Fault Tree Editor menu bar: **File ➔ Open**

  OR Choose the **Open Diagram**  button on the tool bar.

  Double-click on the fault tree filename you wish to edit.

Selecting and Arranging Logic Symbols

- To begin building a fault tree

  Select the desired object button from the tool bar.

  Then move the shape cursor to the desired location with the mouse and click the left mouse button.

  Use the right mouse button to end the selection or menu option.

- To select objects

  Choose the Pick button  to select a single object

  Choose the Text Pick button  to select a text object.

  Then click on the desired object.
  A dashed line will appear surrounding the selected object(s).

- To delete unwanted objects

  Select the object to be deleted (click on the shape with the left mouse button)
  Fault Tree Editor menu bar: **Edit ➔ Delete**
  OR Choose the Delete key.

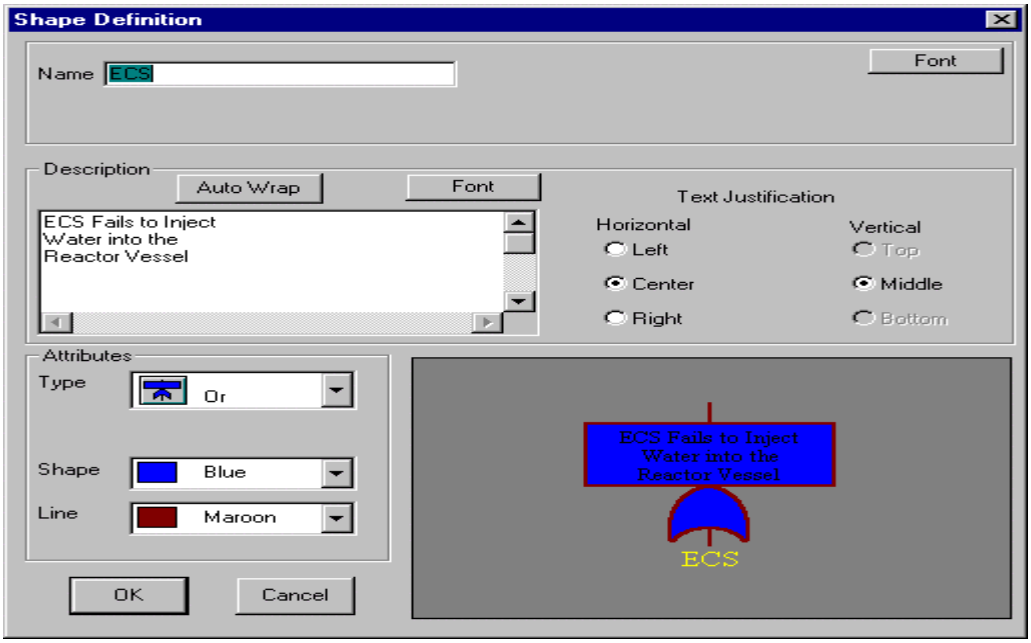
- To move symbols (and their associated text, if any)

  Select the object to be moved
  Drag the object to the desired location.

Naming Gates And Basic Events

- To rename logic or basic event symbols

  Select the desired shape
  Fault Tree Editor menu bar: **Edit ➔ Attributes**
  OR right-click and choose **Edit**,



Then make desired changes.

- To view the current name of a gate or basic event

  Choose the Pick button  to select a single object. On the bottom of the fault tree graphic the basic event name or gate name along with their description will be shown.

Connecting Symbols With Lines

- To link gates and basic events together

  Select the **Line** button 

from the tool bar. Drag from the starting location to the ending location.

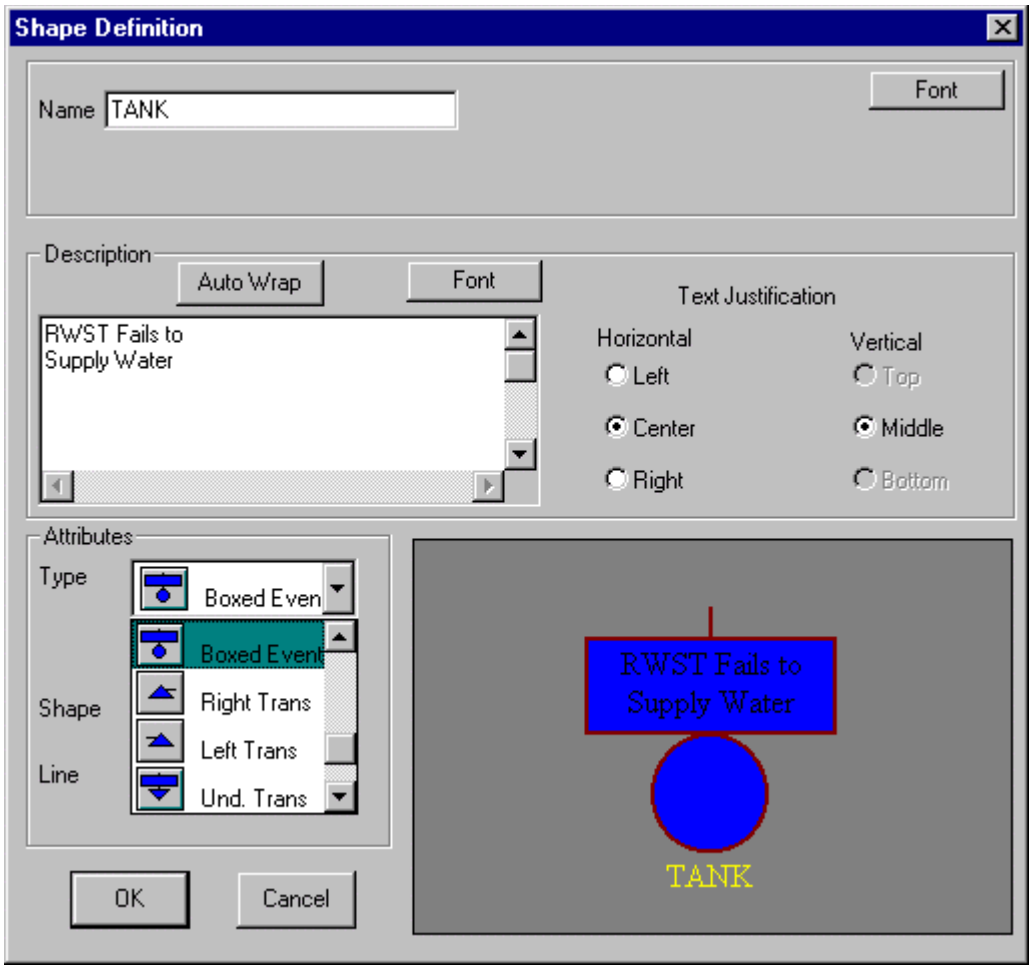
  (Note: if you move symbols after connecting with lines, you may need to redraw the lines unless the lines were moved with the symbols.)

Changing Symbol Type

- To change a symbol (e.g., to change a basic event to an undeveloped transfer)

  Select the shape

  Fault Tree Editor menu bar: **Edit ➔ Attributes**



Choose the **Type** button.

Select the new shape type from the list.

Display Adjustment

- To change the view of the fault tree

  Fault Tree Editor menu bar: **View**

  Zoom allows you to zoom in or out. Or select the button 

  Page Up allows you to move up one page.

  Page Down allows you to move down one page.

  Page Left allows you to move left one page.

  Page Right allows you to move right one page.

  View Normal resets the screen.

  View Fit zooms the fault tree in to fit the screen. Or select the button 

Or

Highlight an area right mouse click and select **zoom selected**.

- To redraw the window

  Fault Tree Editor menu bar: **View ➔ Refresh or F5**

Entering Descriptive Text

- To enter text (for a title or other description)

  Choose the Text Object button  from the tool bar. The cursor will change to the Text cursor  .

  Position the text cursor at the desired location.

  Click the mouse. The *Text Attributes* dialog will be displayed.

- To move the associated symbol only

  Choose the Pick button  from the tool bar. The cursor will change to the Pick cursor  .

- To move text only

  Choose the Text Pick button  from the tool bar. The cursor will change to the Text Pick cursor  .

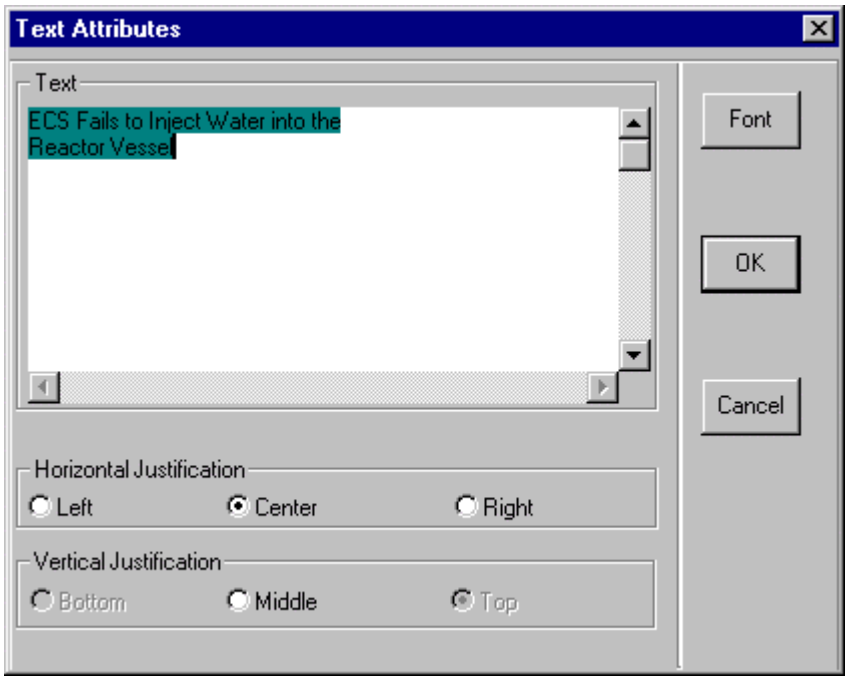
  Select the desired text.

  Drag the selected text to the new location.

- To change existing text and attributes

  Select the desired text using the Text Pick cursor  .

  Right-click and choose **Edit**. The *Text Attributes* dialog will be displayed.



Text - Descriptive text for a shape or explanatory text for the entire fault tree. Maximum of 600 alphanumeric upper- and lowercase characters.

Justification

Horizontal - The horizontal alignment of the text.

Vertical - The vertical alignment of the text.

Text Font - The font size or type for the text and the color of the text.

OK - Close the *Text Attributes* dialog and add the input text to the diagram.

Cancel - Close the *Text Attributes* dialog without adding the text to the diagram.

View Preferences

Fault Tree Editor menu bar: **View ➔ Preferences**
OR right-click and choose **Preferences**

Check boxes under **Other** to do the following:

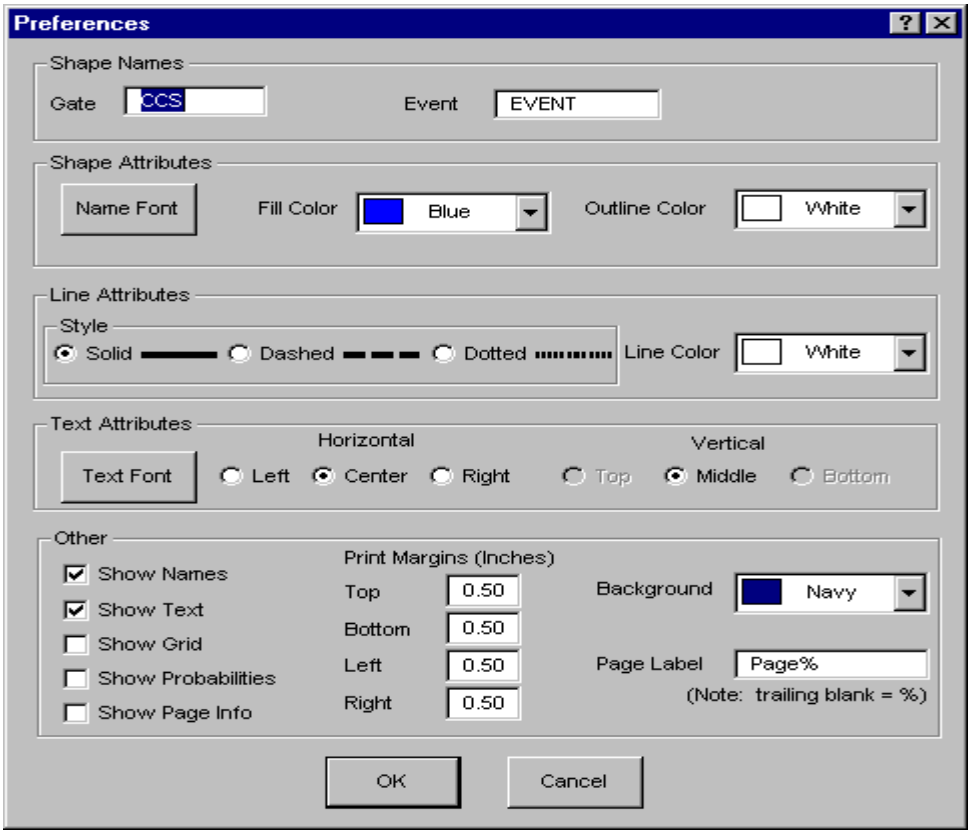
Show Names: This places the event name under the symbol.

Show Text: This will show the event or fault tree text.

Show Grid: This will place grid lines to help in placing symbols.

Show Probabilities: This will put the basic event's probability to the side of the event.

Show Page Info: This will place the fault tree title and the date the fault tree was last modified along with the page number for the fault tree on the bottom of the graphic.



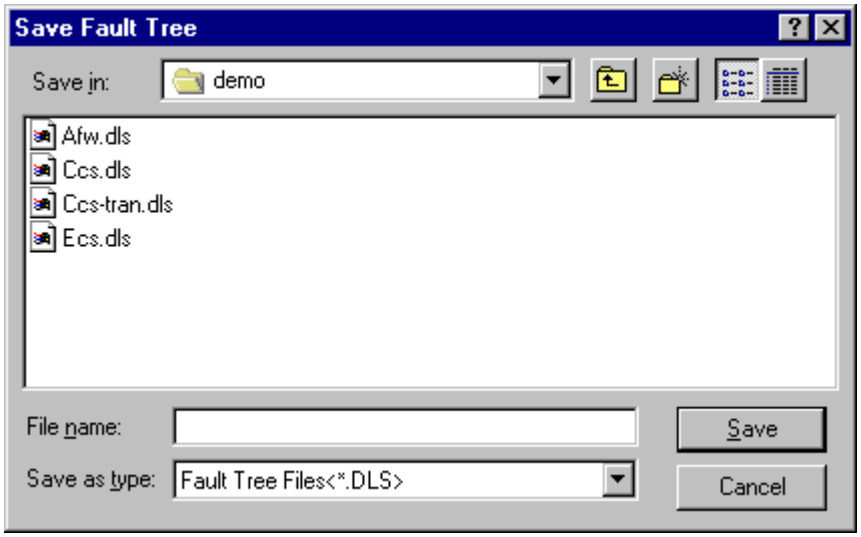
Ending The Editing Session

- To save the fault tree graphic

  Fault Tree Editor menu bar: **File ➔ Save**,

  OR Choose the **Save Diagram** button  on the tool bar

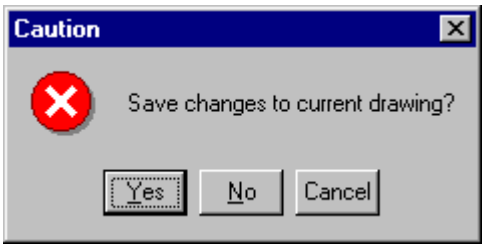
  Name the fault tree file as directed.



- To exit without saving

  Event Tree Editor menu bar: **File ➔ Exit**

  Choose **No** to quit without saving.



3. GENERATING FAULT TREE CUT SETS

Section 3 describes how to generate fault tree cut sets. Model preparation prior to generating cut sets is discussed, and the various analysis and truncation options are described. Cut set display features are also presented.

3.1 Prerequisites for Generating Fault Tree Cut Sets

① Fault tree logic was created by using the fault tree graphics editor, fault tree logic editor, or loaded into the database via the MAR-D interface.

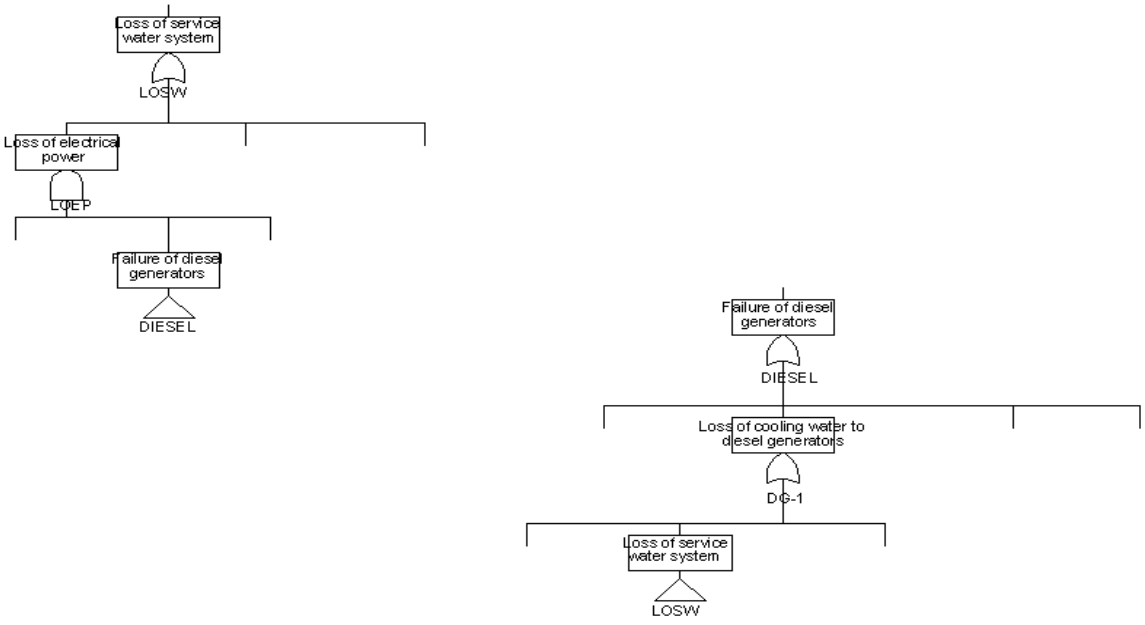
② Basic event data were added through the **Modify ➔ Basic Events** menu.

③ Basic event data was prepared for model processing by using the **Generate** option.

④ Fault tree transfers are properly modeled so that

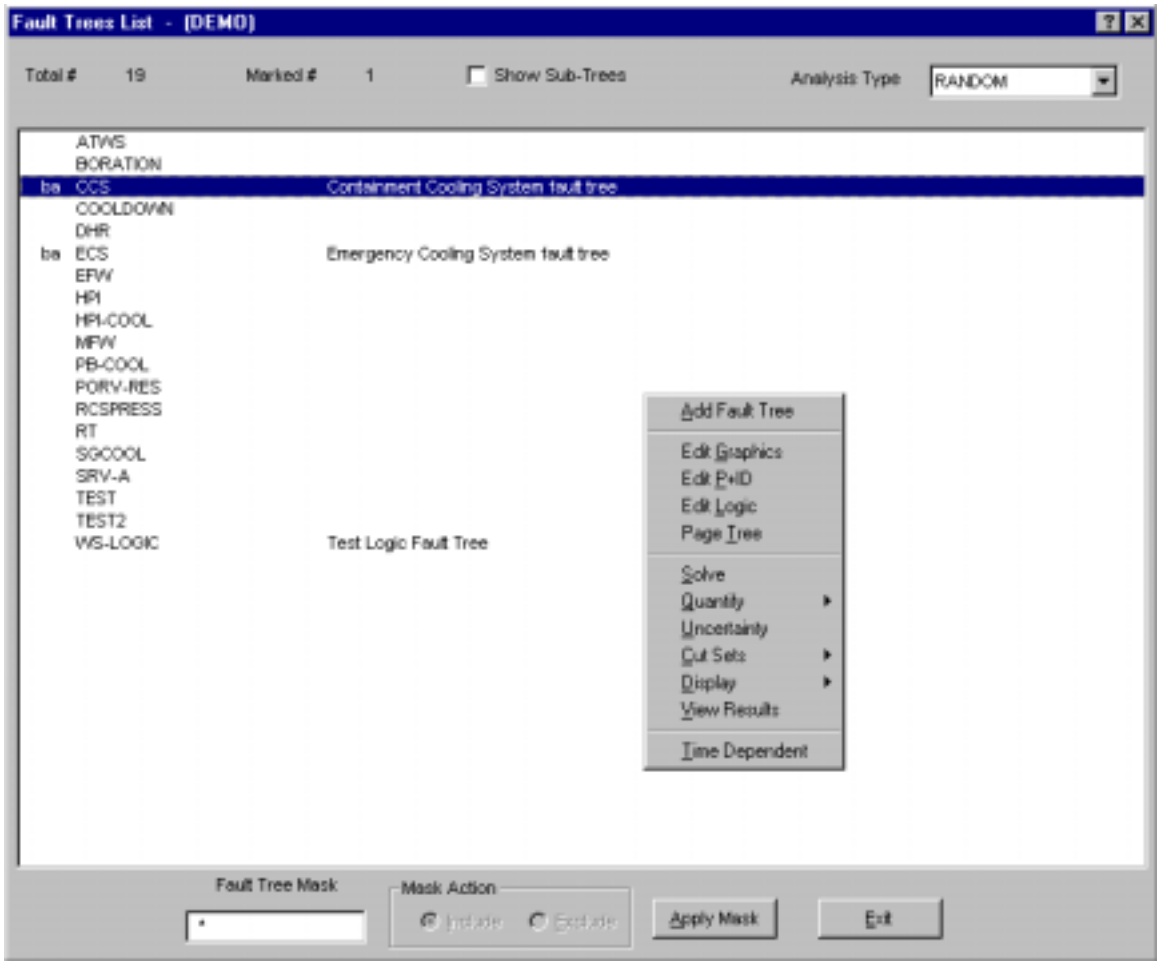
- ◊ There are no logic loops in the fault trees
- ◊ There is only one top gate in each fault tree
- ◊ The naming of transfer gates and fault tree filenames is consistent.

Logic Loop Example



The correct way to "break the loop" will depend on which system is being analyzed.

3.2 Menus and options for fault tree cut set generation

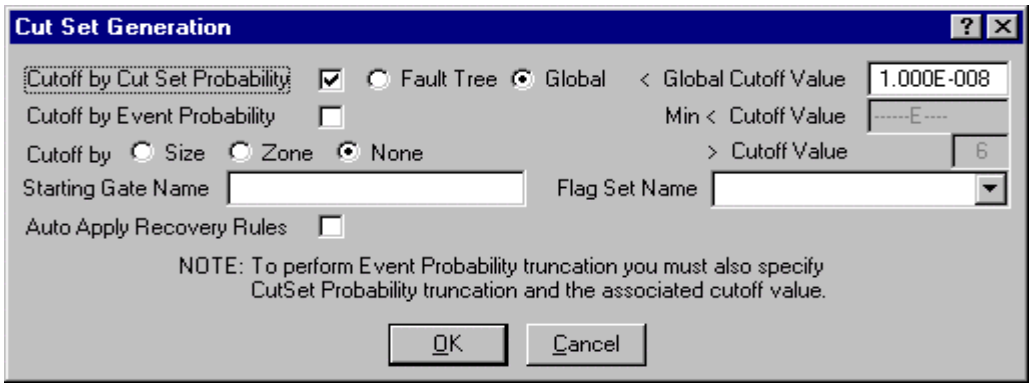


- Select the **Fault Tree** option from the menu.
- Mark the Fault Trees using the mask feature, or using the mouse.
- Right-click to invoke the pop-up menu.
- Select the **Solve** option.

b - flags fault trees with base case cut sets
a - flags fault trees with alternate case cut sets

This option uses the fault tree logic from all fault trees that link to the top gate in the system. The fault tree probability is quantified using the minimal cut set upper bound approximation.

Truncation Parameters



♦ Select the desired truncation parameters on the dialog, and choose **OK** to begin generating cut sets.

- ***Cutoff by Cut Set Probability*** - If you select this check box, then those cut sets below the cutoff value will not be retained. Choose one of the radio buttons:

  Global - Only those cut sets whose product for all of its event probabilities is greater than or equal to the value in the ***< Global Cutoff Value*** field will be kept.

  Fault Tree - Only those cut sets whose product for all of its event probabilities is greater than or equal to the fault tree's cutoff value will be kept. This value is entered in the **Modify ➔ Fault Tree** option.

- ***Cutoff by Event Probability*** - If you select this check box, then you must also select the ***Cutoff by Cut Set Probability*** check box. This option will retain cut sets comprised of basic events that are above the ***Min < Cutoff Value*** even if the cut set is below the ***Global Cutoff Value***.

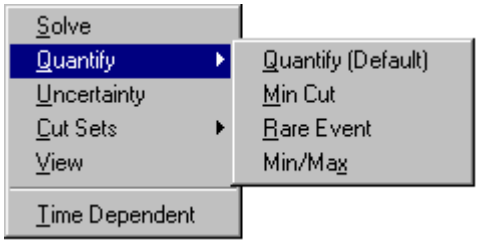
- ***Cut Set Size*** - If you select this check box, then cut sets having more events than specified in the ***> Cutoff Value*** field will not be retained. If you select the ***Zone*** check box, then cut sets having more Zone Flagged Events than specified in the ***> Cutoff Value*** field will not be retained. If neither check box is selected, then the number of events in a cut set will not affect whether the cut set is retained or discarded.

- ***Starting Gate Name*** - If you leave the field blank, the top gate in the system will be used. If you specify a gate, that gate will be used as though it were the top gate.

- ***Flag Set Name*** - If you leave the field blank, the system-specific flag set, if any, will be used. If you specify a flag set, that flag set will be used during processing.

- ***Auto Apply Recovery Rules*** – If you check box, any recovery rules associated with this fault tree will automatically be applied after the fault tree cut sets have been generated.

Quantify



This submenu provides options for requantifying existing current case system cut sets. These options are designed to quickly requantify the cut sets when data changes have been made. (Note: if data changes increase the failure probability of an event, the Solve option should be used instead.)

Minimal Cut Set Upper Bound Approximation

This calculation approximates the probability of the union of the minimal cut sets for the fault trees. The equation for the minimal cut set upper bound is

$$S = 1 - \prod_{i=1}^{m} (1 - C_i)$$

where

S = minimal cut set upper bound for the system unavailability,

Ci = probability of the i th cut set, and

m = the number of cut sets.

Example: If the cut sets for a system are X = A ∪ B∪ C (i.e., the union of three events, A, B, and C); then the cut sets can be written as X = A + B + C with the plus symbol indicating union. The system unavailability computed from the minimal cut set upper bound approximation is then X = 1 - (1 - A)(1 - B)(1 - C).

Min Max Quantification

The Min-Max quantification option quantifies the current case cut sets using the exact probability quantification algorithm. From the example above, the exact system unavailability is

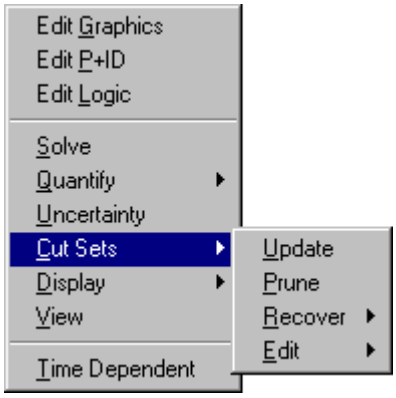
X = (A + B + C) - (A * B + A * C + B * C) + (A * B * C),

with the number of passes in this example being 3, corresponding to the number of pairs of parentheses.

Uncertainty

Performs Monte Carlo or Latin Hypercube uncertainty analysis for the selected fault tree. Fault tree uncertainty analysis is discussed further in Section 9.

Cut Set Analysis



This sub-menu provides options for cut set manipulation.

Cut Set Update

This option uses the existing current case cut sets (unless the user specifies that base case cut sets are to be used instead). Non-minimal cut sets are eliminated and the fault tree probability is quantified using the minimal cut set upper bound approximation.

Prune Cut Sets

This is one of the options that allows you to eliminate cut sets from a selected fault tree that contains events which conflict in some way with one another. For example, if technical specifications prohibit two pumps from being unavailable due to test and maintenance, a cut set containing test and maintenance events for the two pumps would be invalid.

Analysis Type

Select the RANDOM analysis type for material covered in this class. The other analysis types are provided for performing fire, flood, seismic, and other hazard analyses.

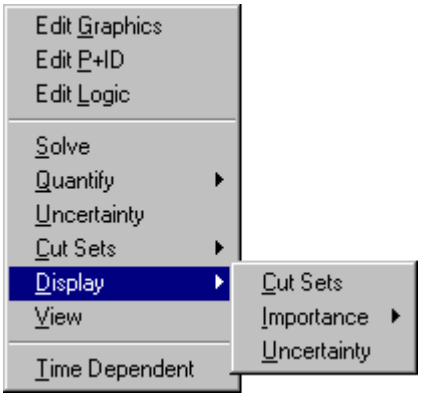
Recover Option

This option allows for post-processing rules to be created and applied to the fault tree cut sets. The rules can be directed at either a single fault tree or all fault trees.

Edit Option

This option provides a mechanism to manually edit the fault tree cut sets.

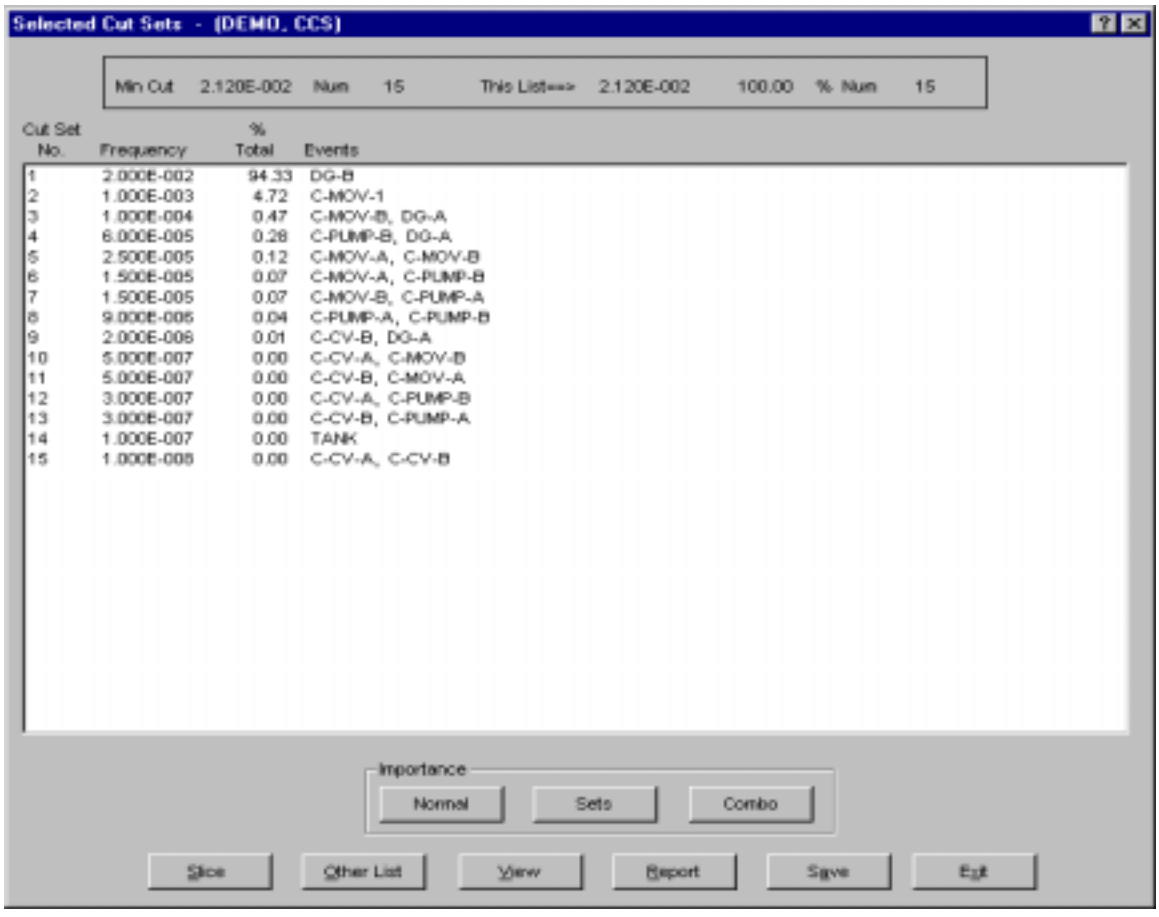
Displaying Fault Tree Cut Set Results



- To display fault tree cut sets, select **Fault Tree** from the menu bar.
- Highlight the fault tree that you want to view; right-click to invoke the pop-up menu.
- Select the **Display** ➔ **Cut Sets** option.

Selecting Fault Tree Cut Sets to View

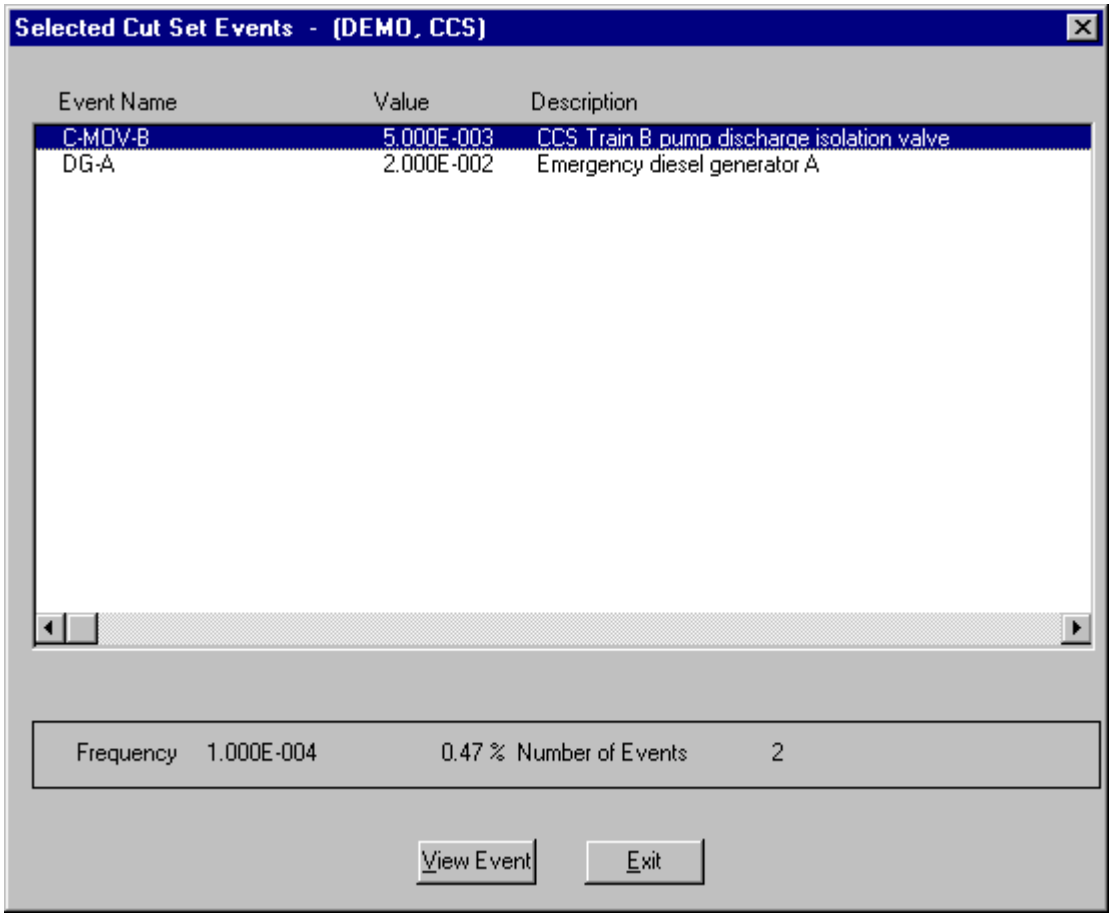
- The fault tree cut sets and minimal cut set upper bound approximation of the fault tree failure probability are now displayed.



- To view the basic events in a cut set, highlight the cut set, choose the **View Cut Set** button.

Viewing Fault Tree Cut Sets

- The basic events, their failure probabilities and descriptions are now displayed.

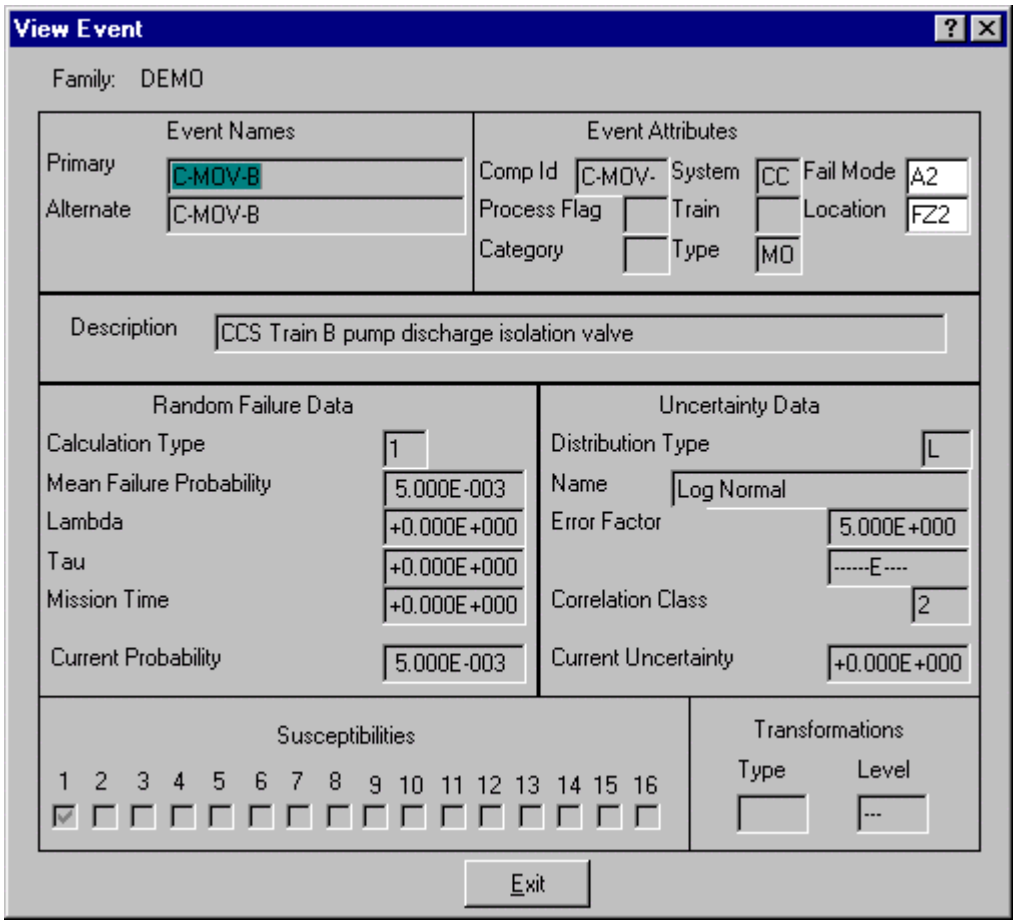


- To view individual basic event information, highlight the basic event, and choose the **View Event** button.

- To find out where the events in the cut set came from, right-click on a cut set and select the **Path Search** option.

The Path Search option traces through the fault tree logic to indicate exactly where in the fault tree the cut set came from.

Viewing Basic Event Information for an Individual Cut Set



- The individual basic event information is now displayed.